



TITLE:

置換群の共役群計算の高速化 (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

宮本, 泉

CITATION:

宮本, 泉. 置換群の共役群計算の高速化 (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2012, 1814: 36-42

ISSUE DATE:

2012-10

URL:

<http://hdl.handle.net/2433/194554>

RIGHT:

置換群の共役群計算の高速化

宮本 泉*

IZUMI MIYAMOTO

山梨大学

UNIVERSITY OF YAMANASHI

1 正規化群と部分群の共役

G : 集合 Ω 上の置換群、 $H, K: G$ の部分群

- H の G における正規化群 $\text{Norm}(G, H) = \{g \in G \mid g^{-1}Hg = H\}$
- H と K は G において共役 $g^{-1}Hg = K$ for some $g \in G$

Cannon-Holt: “The transitive groups of degree 32” (2008.12) にある記述

Computing normalisers and testing conjugacy of subgroups are notoriously difficult problems even in permutation groups of small degree.

GAP システムによる計算実験 ($G = \text{Sym}(n)$ のとき)

$n = 31$ 次まで。GAP-ライブラリの可移置換群リストを使う。

H : 可移置換群

- 正規化群計算: 少ないが無視できない個数の H で困難
- 部分群の共役計算: 困難な H と g はほとんど無い

可移置換群 (の同型類) の個数

31 次まで:	40 238 個
32 次	: 2 801 324 個

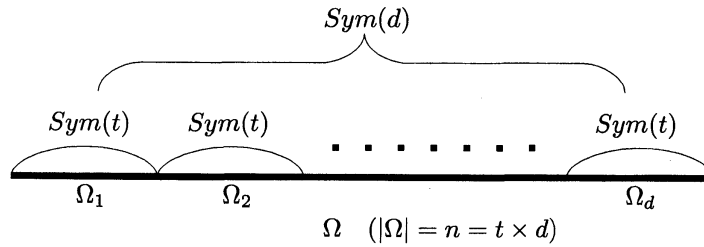
32 次の場合すべての可移群のリストは利用できていない。

- 部分群の共役計算で困難な場合が、少なからず出てくる。

*imiyamoto@yamanashi.ac.jp

2 Block System の利用

Block System と WreathProduct



$$Sym(t) \wr Sym(d) = \text{WreathProduct}(Sym(t), Sym(d))$$

$\{\Omega_1, \Omega_2, \dots, \Omega_d\}$: ブロックシステム

$H \subseteq W = Sym(\Omega_1) \wr Sym(d)$, $K \subseteq W' = Sym(\Omega'_1) \wr Sym(d)$ のとき

Step 1. $g^{-1}Wg = W' \iff \{\Omega_1^g, \Omega_2^g, \dots, \Omega_d^g\} = \{\Omega'_1, \Omega'_2, \dots, \Omega'_d\}$

Step 2. $\exists? h \in W'$ such that $h^{-1}g^{-1}Hgh = K$

定義: H は非原始的 $\iff H$ は可移、 $H \subseteq Sym(t) \wr Sym(d)$

3 2 次の可移群

原始的 7 個

非原始的 2 801 317 個

ブロックシステムで作用を分解して共役をとった上で、特別な デザイン を利用する。(Cannon-Holt)

1. $Ker(H) = H \cap Sym(t)^d$... ブロックシステムの固定部分群

固定点を使った "デザイン" の同型計算 (Leon) を利用

2. $Im(H) = H Sym(t)^d / Sym(t)^d$... ブロックシステム上 d 次の置換群

3. $\exists? g \in \text{デザイン}$ の自己同型群 $\cap \text{Prelimage}(\text{Norm}(Sym(d), Im(H)))$ such that $g^{-1}Hg = K$

$|\Omega_1| = 2$ のとき使用。 $H \subseteq W \cap W'$ のなる場合あり。Cannon-Holt は概説のみの論文で、デザイン の詳細も不明。

3 アソシエーションスキーム

可移置換群からつくるアソシエーションスキーム $(\Omega, \{R_i\}_{i=0,1,2,\dots,d})$

R_i = 群 H の $\Omega \times \Omega$ への作用の orbit

【例】 $\Omega = \{1, 2, 3, 4, 5, 6\}$, $H = \text{Group}((1, 5, 4)(2, 6, 3), (1, 6, 3, 2, 5, 4))$

$$\left. \begin{aligned} R_0 &= \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}, \\ R_1 &= \{(1, 2), (2, 1), (3, 4), (4, 3), (5, 6), (6, 5)\}, \\ R_2 &= \{(1, 3), (1, 4), (2, 3), (2, 4), \dots, (6, 1), (6, 2)\}, \\ R_3 &= {}^tR_2 = \{(3, 1), (4, 1), (3, 2), \dots, (1, 6), (2, 6)\} \end{aligned} \right\} \begin{array}{l} \Omega \times \Omega \text{ 上の orbit} \\ 4 \text{ 個} \end{array}$$

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 2 & 3 & 3 \\ 1 & 0 & 2 & 2 & 3 & 3 \\ 3 & 3 & 0 & 1 & 2 & 2 \\ 3 & 3 & 1 & 0 & 2 & 2 \\ 2 & 2 & 3 & 3 & 0 & 1 \\ 2 & 2 & 3 & 3 & 1 & 0 \end{pmatrix} \end{matrix}$$

アソシエーションスキームの同型と共役計算

【例 (続き)】 自己同型群

$$\text{Aut}(A) = \text{Group}((3, 5, 4, 6), (1, 6, 2, 5)(3, 4)) \supsetneq H$$

$\text{Aut}(A)$ の作用

$$\begin{matrix} \circlearrowleft & \circlearrowleft \\ R_0, & R_1, & R_2 & \rightleftharpoons & R_3 \end{matrix}$$

命題 3.1

$$\text{Norm}(H) \subseteq \text{Aut}(A)$$

可移群 $H, K \subseteq \text{Sym}(n)$ の共役計算 へのアソシエーションスキームの利用

(これだけでは、あまり効果はない)

$A, B : H, K$ が作るアソシエーションスキーム

- $\exists?$ 同型 g such that $g^{-1}Ag = B$
- $\exists? h \in \text{Aut}(B)$ such that $h^{-1}g^{-1}Hgh = K$

4 アルゴリズム

$H, K : \text{可移群} \subseteq \text{Sym}(n)$ 同じアソシエーションスキーム A を作る様に共役をとっておく。

$G = \text{Aut}(A) \subseteq \text{Sym}(n)$ で、 $g \in G$ such that $g^{-1}Hg = K$ を探索するとき、

1. H は可移 $\implies \exists h \in H$ such that $(1^h)^g = 1 \in \Omega$ より $g \in G_1 = \{x \in G | 1^x = 1\}$ (点 1 の固定部分群) としてよい。
このとき、 $g^{-1}H_1g = K_1$ も成立する。
2. さらに、 G_1 と H_1 が同じ orbit O_1 をもつときは、 $\exists h_1 \in H_1$ such that $O_1 \ni (2^{h_1})^g = 2$ より、
 $g \in G_{1,2} = \{x \in G_1 | 2^x = 2\}$ としてよい。
このとき、 $g^{-1}H_{1,2}g = K_{1,2}$ も成立する。
3. 以下、可能ならば、同様に繰り返す。

$G = \text{Aut}(A)$ と制限されているので、繰返せる可能性が高い。

このような条件を満たす g が無ければ、 H と K は共役にならない。

簡単な例

H, K が 2 重可移の場合 (\Rightarrow ブロックシステムは存在しない)

(H が 2 重可移 \iff 可移、かつ、1 点固定部分群 $H_1 = \{h \in H | 1^h = 1\}$ は $\Omega \setminus \{1\}$ 上可移)

【例】 $H = \text{PrimitiveGroup}(31, 10) = L(5, 2)$

1. $H, \text{Sym}(n)$ は可移 $\Rightarrow g \in \text{Sym}(n)$ such that $H^g = K$ は、 $g \in \text{Sym}(n-1)$ で探索してよい。

2. $g' \in \text{Sym}(n-1)$ such that $g'^{-1}H_1g' = K_1$ を探索。(0.2 秒)

3. $H_1, \text{Sym}(n-1)$ は $\Omega \setminus \{1\}$ 上可移 $\Rightarrow g \in \text{Sym}(n-2)$ で探索してよい。

4. $g^{-1}(g'^{-1}H_1g')_2g = K_{1,2} \Rightarrow g \in \text{Norm}(\text{Sym}(n-2), K_{1,2})$ で探索。(Norm - 4.4 秒、 g - 0.0 秒)

$\Rightarrow (g'g)^{-1}H(g'g) = K$

直接計算: $\text{RepresentativeAction}(\text{Sym}(31), H, K)$ (10 時間より大)

4.1 プログラム

2つのプログラムを作成した。

- ConjAS ... H と K が同じアソシエーションスキームを作るように共役を計算。
アソシエーションスキームの自己同型群 G で、前述のアルゴリズムによって、 H を K に移す共役元を計算。
- ConjASB... ConjAB を使うときブロックシステムが利用できる場合は、ブロック上の作用 $\text{Im}(H)$ と $\text{Im}(K)$ の共役を $\text{Im}(G)$ で計算。
その後、 $\text{Ker}(H)$ を $\text{Ker}(K)$ に移す操作は抜きで、
 $\text{Prelm}(\text{Norm}(\text{Im}(G), \text{Im}(H))) \subseteq G$ で H を K に移す共役元を計算。

5 実験結果

3 2 次のアソシエーションスキームの分類リスト <http://kissme.shinshu-u.ac.jp/as/> を使ってできる可移な置換群を利用する。

Example 5.1 $as32[38]$ を作るいくつかの可移置換群

ブロックシステム $[[1, 2, 3, 4], [5, 6, 7, 8], \dots, [29, 30, 31, 32]]$

種類分け

$\left. \begin{array}{l} \text{ブロック上の作用 } \text{Im}(H) (8 \text{ 次の群}) \\ \text{ブロックを固定する部分群 } \text{Ker}(H) \end{array} \right\} \text{ が同一な群 : } \begin{array}{l} 309 \text{ 種類} \\ 1109 \text{ 個} \end{array}$

$G = \text{Aut}(as32[38])$

$N = \text{Norm}(G, \text{Ker}(H)) \cap \text{Prelm}(\text{Norm}(\text{Im}(G), \text{Im}(H)))$

$as32[38]$ を作るいくつかの可移置換群 (続き)

$\text{Im}(H) = \text{Im}(K)$, $\text{Ker}(H) = \text{Ker}(K)$ となる H と K の共役計算

1. RepresentativeAction(N, H, K) ... GAP の共役元計算の関数
2. ConjAS(G, H, K)
3. ConjASB(G, H, K)
4. RepresentativeAction(G, H, K)

計算時間 上の条件を満たす H, K , 2254 組すべての合計 (ミリ秒)

	共役計算時間	N の計算時間	合計計算時間
1.	1079270	1986138	3065408
2.	8982993		8982993
3.	3189934		3189934

4. 30 分以上かかる場合が多くでてきて、計算を断念

共役計算時間の分布

1. 8.0 秒, 7.9 秒, 4.3 秒, 4.3 秒, ..., 4.0 秒 (4.0 秒以上、38 個)
2. 64 秒 (2 個), 49 秒 (2 個), 29 秒, 19 秒, ... (10 秒以上、33 個)
3. 4.0 秒, 4.0 秒, 4.0 秒, 4.0 秒, ..., 3.8 秒 (3.8 秒以上、53 個)

Example 5.2 $as32[3]$ を作る 2 組 4 個の位数 11010048 の可移置換群

$G = \text{Aut}(as32[3])$

ブロックシステム $[[1, 2, 3, 4], [5, 6, 7, 8], \dots, [29, 30, 31, 32]]$

$\text{Im}(H) = \text{Im}(K)$ (位数)	TransitiveGroup(8, 36) 168	TransitiveGroup(8, 37) 168
$\text{Ker}(H) = \text{Ker}(K)$	基本可換 2 群、位数 65536	
G の位数	4438236667576320	4438236667576320
N の位数	18492652781568	36985305563136
計算時間 (ミリ秒)	1. 65531	1. 165396
	2. 14179	2. 13793
	3. 2879	3. 2198

$as32[3]$ を作る 2 組 4 個の位数 11010048 の可移置換群 (続き)

ConjAB におけるアルゴリズムの適用状況

1. H は可移 $\implies \exists h \in H$ such that $(1^h)^g = 1 \in \Omega$ より $g \in G_1 = \{x \in G | 1^x = 1\}$ (点 1 の固定部分群) とできる。
 $|G : G_1| = 32$ このとき、 $g^{-1}H_1g = K_1$ も成立する。
2. G_1 と H_1 が同じ orbit O_1 をもち、 $|O_1| = 28$ で、 $\exists h_1 \in H_1$ such that $O_1 \ni (2^{h_1})^g = 2$ より、
 $g \in G_{1,2} = \{x \in G_1 | 2^x = 2\}$ とできる。
 $|G_1 : G_{1,2}| = 28$ このとき、 $g^{-1}H_{1,2}g = K_{1,2}$ も成立する。

3. $G_{1,2}$ と $H_{1,2}$ が同じ orbit O_2 をもち、 $|O_2| = 3$ で、 $\exists h_2 \in H_{1,2}$ such that $O_2 \ni (3^{h_2})^g = 3$ より、 $g \in G_{1,2,3} = \{x \in G_{1,2} | 3^x = 3\}$ とでき、ここで探索する。
 $|G_{1,2} : G_{1,2,3}| = 3$ このとき、 $g^{-1}H_{1,2,3}g = K_{1,2,3}$ も成立する。

$\Rightarrow G_{1,2,3}$ の位数 = 1651129712640 (G の位数の $32 \times 28 \times 3$ 分の 1)
 (N の位数 = 18492652781568)

計算時間の平均

GAP を使って生成できた 32 次の可移置換群：約 27 万個

$g \in \text{Random}(G)$, $G = \text{Aut}(as(H))$ として、 $\text{ConjAS}(G, H, g^{-1}Hg)$ を計算

assoc. scheme	群の個数	総計算時間	平均計算時間
全体 3920 個	270995	364979 秒	1.35 秒
$as32[4182]$	118	39499 秒	335 秒 *

*：平均計算時間で最も困難だった場合

$as[4182]$ からできる 24 番目の群の共役計算 15817 秒 平均計算時間が予想外に速い

- 同様なアルゴリズムによる正規化群計算時間 2 秒程度以上が多い
- $\text{Index } |G : \text{Norm}(G, H)| \leq 5000$ のときは、このコセットの代表から共役元を直接計算している (ただの逐次探索)
- 27 万個：32 次の可移群訳 280 万個の 1/10 程度の個数
 → 作れたのは (共役計算が) 簡単な群ばかりかも知れない。

6 問題点

$\text{RepresentativeAction}(G, H, K)$ は、ほとんどの場合、1 秒以内に計算できる。一方で、10 時間でも計算できない場合もある。ブロックシステムを利用しないで、 G を使うと、

Example ?? では、30 分以上かかる場合も多い。

Example ?? では、10 時間かけても計算できなかった。

$\text{RepresentativeAction}(N, H, K)$ と $\text{RepresentativeAction}(G, H, K)$ の違い

(Cannon-Holt の論文と比較のため) ブロックシステムを利用して、 G で $\text{Ker}(H)$ と $\text{Ker}(K)$ の共役の計算、その後 $N = \text{Norm}(\text{PrelImage}(\text{Norm}(\text{Im}(G), \text{Im}(H))), \text{Ker}(H))$ の計算 (手間が面倒、また、計算時間がかかることも多い) → 実験対象の準備が困難 (紹介できる実験結果が少ない)

紹介しているアルゴリズムの特徴

- ブロックシステムが利用できない場合でも利用可能な場合がある。
 - ブロックシステムのアルゴリズムと併用も可能。(実現できていない)
 - 使える条件が複数あり、使うと高速化できることもあるが、逆効果になることもある。
1. H は可移 $\Rightarrow \exists h \in H$ such that $(1^h)^g = 1 \in \Omega$ より $g \in G_1 = \{x \in G | 1^x = 1\}$ (点 1 の固定部分群) としてよい。
 このとき、 $g^{-1}H_1g = K_1$ も成立する。

2. さらに、 G_1 と H_1 が同じ orbit O_1 をもつときは、 $\exists h_1 \in H_1$ such that $O_1 \ni (2^{h_1})^g = 2$ より、 $g \in G_{1,2} = \{x \in G_1 | 2^x = 2\}$ としてよい。
このとき、 $g^{-1}H_{1,2}g = K_{1,2}$ も成立する。
3. 以下、可能ならば、同様に繰り返す。
 $G = \text{Aut}(A)$ と制限して、繰り返せる可能性を高くしている。